**API Covoiturage**

* symfony new ApiCovoiturage
* ------------- se positioner dans le dossier d’application -------------------
* cd APICovoiturage
* ----------- install dependencies via composer --------------
* composer install
* symfony server:start

On peut lister toutes les variables d’environnement grâce à cette commande sur le terminal (CTRL -U

php bin/console debug:container --env-vars

maker bundle for make: commands

composer require symfony/maker-bundle –dev

ORM installation:

Composer require orm

**JWT Authentication token**

in php.ini faut decommeter:

extension=sodium

composer require lexik/jwt-authentication-bundle:^3.x-dev

Creation de la classe Compte

Dans la methode login, un JWT token est generé a chaque reprise, avec un duré de vie de 24 h ( configuration dans lexik\_jwt\_authentication.yaml, token\_ttl: 86400).

Creation des entites : Eleve, Voiture, Trajet, Ville, Marque,

**Mise a jour d’etudiant**

#[Route('/updateStudent/{id}/{firstname}/{lastname}/{phone}/{email}/{brand}/{model}/{seatsnumber}', name: 'app\_student\_update', methods: ['PUT'])]

public function updateStudent(Request $request, EntityManagerInterface $entityManager): JsonResponse

{

try {

$id = $request->attributes->get('id');

$student = $entityManager->getRepository(Student::class)->find($id);

if (!$student) {

throw new Exception('Student not found');

}

$firstname = $request->attributes->get('firstname');

$lastname = $request->attributes->get('lastname');

$phone = $request->attributes->get('phone');

$email = $request->attributes->get('email');

$brand = $request->attributes->get('brand');

$model = $request->attributes->get('model');

$seatsnumber = $request->attributes->get('seatsnumber');

$student->setFirstName($firstname);

$student->setLastName($lastname);

$student->setPhone($phone);

$student->setEmail($email);

$car = $student->getOwns();

if (!$car) {

$car = new Car();

$student->setOwns($car);

}

$brandEntity = $entityManager->getRepository(Brand::class)->findOneBy(['carBrand' => $brand]);

if (!$brandEntity) {

$brandEntity = new Brand();

$brandEntity->setCarBrand($brand);

$entityManager->persist($brandEntity);

}

$car->setModel($model);

$car->setSeatsNumber($seatsnumber);

$car->setIdentify($brandEntity);

$entityManager->persist($car);

$entityManager->persist($student);

$entityManager->flush();

return new JsonResponse(['message' => 'Student successfully updated']);

} catch (Exception $e) {

return new JsonResponse(['error' => $e->getMessage()], 400);

}

}

**Liste les Eleves**

#[Route('/findAllStudents', name: 'app\_student\_index', methods: ['GET'])]

public function index(Request $request, EntityManagerInterface $entityManager): JsonResponse

{

try {

$studentRepository = $entityManager->getRepository(Student::class);

$students = $studentRepository->findAll();

if (!$students) {

throw new \Exception('No students found');

}

$studentData = array\_map(function (Student $student) {

return [

'id' => $student->getId(),

'firstname' => $student->getFirstName(),

'lastname' => $student->getLastName(),

'email' => $student->getEmail(),

'phone' => $student->getPhone(),

'city' => $student->getLives()->getCityName(),

'zip\_code' => $student->getLives()->getZipCode(),

'car' => $student->getOwns()?->getModel(),

'seats' => $student->getOwns()?->getSeatsNumber(),

'registration' => $student->getOwns()?->getRegistrationNumber(),

'brand' => $student->getOwns() && $student->getOwns()->getIdentify() ? $student->getOwns()->getIdentify()->getCarBrand() : null,

];

}, $students);

return new JsonResponse($studentData);

} catch (\Exception $e) {

return new JsonResponse(['error' => $e->getMessage()], 400);

}

}

To authenticate a user for a request, you need to include the JWT token in the `Authorization` header of the request. The token should be prefixed with `Bearer `.

Here's an example of how you can do this using `curl`:

```bash

curl -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE3MDk0NTAwODEsImV4cCI6MTcwOTUzNjQ4MSwicm9sZXMiOlsiUk9MRV9VU0VSIl0sInVzZXJuYW1lIjoibG9naW4xIn0.KPTF8G3X56q46JGQqfecwtpfKFqXi\_d0E9dIR-K\_Oz9j5jC7z5Ay7BrXEHX8vqk4DDUKGXA0cGOjNy-z7yHq79KVy5lUjyrtQL2gpBE997rQf9OA5d-nYqFskABvo8vXYCQrNFCXflE63JXp1mZ94F2KifsIh-crvk0SJfZEIL-AOFQRdcqe6GQAnc1qTvAXFdAvhQTXvpH1os\_b4\_26ZI0gFFf1e\_B9BdFVD4dTq8x8\_kE16RVySHCX0wGD0NX6YfNOSMir7JRIJL3OgK4aCbei4IP6tlTuXXmGdlfXyX\_eOUgFF5Ne\_CwmyASUXrWLUlrKLt0BbXvekyADmH699wdAXBCcg7hQpQGLbzdf8XCe42CnmMOks930FI5KfvZPz2-T\_nUedWPJBAHiHcHM8eOit3C6MqLcaemR88T-kPIffclZOhtFxuDdWhoa2H\_qI49g1Le4XNO8KKKvZ\_7C4gU-mWv5gH4damNThKLLhaaOpezfh2fmkvHTEh6QVcgIGqjf-WLOYWjPgfq\_O9SqaTGWXoNXdiOZB5ArSSIJPE3ti96iafmMbH6a5Fwav153cAATxH7tWTLl24RB\_hsleUIDZ9zZ62GA-kXT0D4FqAxTHNZ7h2gCnfwV8GhSATo5lZ4\_JZM\_2Roe2vYRFVlQc\_k1okjnpQ609XqCcMLSNxc" http://127.0.0.1:8000/insertPersonne/Milan,Milanovic,12243556,mjovanovic@gmail.com,Vannes,Clio

```

In the above command, `-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE3MDk0NTAwODEsImV4cCI6MTcwOTUzNlQ4MSwicm9sZXMiOlsiUk9MRV9VU0VSIl0sInVzZXJuYW1lIjoibG9naW4xIn0.KPTF8G3X56q46JGQqfecwtpfKFqXi\_d0E9dIR-K\_Oz9j5jC7z5Ay7BrXEHX8vqk4DDUKGXA0cGOjNy-z7yHq79KVy5lUjyrtQL2gpBE997rQf9OA5d-nYqFskABvo8vXYCQrNFCXflE63JXp1mZ94F2KifsIh-crvk0SJfZEIL-AOFQRdcqe6GQAnc1qTvAXFdAvhQTXvpH1os\_b4\_26ZI0gFFf1e\_B9BdFVD4dTq8x8\_kE16RVySHCX0wGD0NX6YfNOSMir7JRIJL3OgK4aCbei4IP6tlTuXXmGdlfXyX\_eOUgFF5Ne\_CwmyASUXrWLUlrKLt0BbXvekyADmH699wdAXBCcg7hQpQGLbzdf8XCe42CnmMOks930FI5KfvZPz2-T\_nUedWPJBAHiHcHM8eOit3C6MqLcaemR88T-kPIffclZOhtFxuDdWhoa2H\_qI49g1Le4XNO8KKKvZ\_7C4gU-mWv5gH4damNThKLLhaaOpezfh2fmkvHTEh6QVcgIGqjf-WLOYWjPgfq\_O9SqaTGWXoNXdiOZB5ArSSIJPE3ti96iafmMbH6a5Fwav153cAATxH7tWTLl24RB\_hsleUIDZ9zZ62GA-kXT0D4FqAxTHNZ7h2gCnfwV8GhSATo5lZ4\_JZM\_2Roe2vYRFVlQc\_k1okjnpQ609XqCcMLSNxc"` sets the `Authorization` header to `Bearer ` followed by your JWT token.

Please replace `http://127.0.0.1:8000/insertPersonne/Milan,Milanovic,12243556,mjovanovic@gmail.com,Vannes,Clio` with the actual URL you want to access.

This command will send a request to the specified URL with the JWT token in the `Authorization` header. The server will check the token, and if it's valid, it will authenticate the user for the request.

write me a curl command to send this 127.0.0.1:8000/insertPersonne/Milan,Milanovic,12243556,mjovanovic@gmail.com,1,1 using this jwt token "token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE3MDk0NTMwNzIsImV4cCI6MTcwOTUzOTQ3Miwicm9sZXMiOlsiUk9MRV9VU0VSIl0sInVzZ XJuYW1lIjoibG9naW4xIn0.S1hln3xx96owajF\_f12DjFsi-ZWsxTZAkdpAYAz9Fys2TSqInBarANhnZ-1hp3HRF1FXSikYFd5qumxNKJOT-OPfAaUW0LvafOzSKAUAiYJYmtippGy-leQTODMwcdxrkEy3pW5 KUzFoc0R86J95t-8fa3-TmXr4zXoT36OTuDukXLWgejc0QFipmSMIWIvUJuF-aaZmslTikGrQTpvpc7zLEo2TMy0HEaFL3y5\_ylP978uZKG46pJS5YUuDZTt-lW4-jj5zMx8M2hevA0pBQz8RSRfKalbaHCmDi gEV-2xNS4H2weWF08JjXIIa8GZtSncOdmEqRC2DMMiFzWbCBv9OlFbl1zadi2eXz0vTEyVFMDUSzm-I9CHKbhQwSAJxP0yxcr\_SuA6s7xb5RXuzJFf5nCUJZx5N95Hj2fxJYgyy1cqJm1Ya2q7S3F6KMF8fiG0 pE47iezbS6Ma2nmggm2F696r6YiOiR-JrQeHoky-mI8H

curl -X POST -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE3MDk0NTMwNzIsImV4cCI6MTcwOTUzOTQ3Miwicm9sZXMiOlsiUk9MRV9VU0VSIl0sInVzZ XJuYW1lIjoibG9naW4xIn0.S1hln3xx96owajF\_f12DjFsi-ZWsxTZAkdpAYAz9Fys2TSqInBarANhnZ-1hp3HRF1FXSikYFd5qumxNKJOT-OPfAaUW0LvafOzSKAUAiYJYmtippGy-leQTODMwcdxrkEy3pW5 KUzFoc0R86J95t-8fa3-TmXr4zXoT36OTuDukXLWgejc0QFipmSMIWIvUJuF-aaZmslTikGrQTpvpc7zLEo2TMy0HEaFL3y5\_ylP978uZKG46pJS5YUuDZTt-lW4-jj5zMx8M2hevA0pBQz8RSRfKalbaHCmDi gEV-2xNS4H2weWF08JjXIIa8GZtSncOdmEqRC2DMMiFzWbCBv9OlFbl1zadi2eXz0vTEyVFMDUSzm-I9CHKbhQwSAJxP0yxcr\_SuA6s7xb5RXuzJFf5nCUJZx5N95Hj2fxJYgyy1cqJm1Ya2q7S3F6KMF8fiG0 pE47iezbS6Ma2nmggm2F696r6YiOiR-JrQeHoky-mI8H" http://127.0.0.1:8000/insertPersonne/Milan,Milanovic,12243556,mjovanovic@gmail.com,1,1